9. Right Hand

Program Name: RightHand.java   Input File: righthand.dat

Most everyone knows about the right-hand rule as a maze escape strategy, or at least they should. In case you don't, it goes like this: if you hold your right hand against the wall to your right at all times, and move through the maze, you shall eventually find your way out – assuming that the maze isn't an infinite trap with no way out.

We've designed a few mazes, but don't necessarily know if they're solvable or not. We suppose it's best to figure out if someone can get out using the right-hand rule after putting them in, so it's your job to let us know!

You will always begin at the top left corner, which is not a wall, and attempt to find your way to the only exit located in the bottom right corner, using the right hand rule. Except for the edges of the bottom right corner exit location, all outside edges of the matrix are walls.

Input

The first integer T will represent how many mazes there are to follow. For each maze, there will be a line containing a single number `n`, which represents both the number of rows and columns of the subsequent square matrix. The next n lines each contain a row of length n. Each character in each row is either a `'.'` or a `'#'`. All `'#'`'s are interior walls, you are only allowed to walk on `'.'`s, and can only move up, down, left, or right – no diagonal movement is possible – the tiny diagonal gap is way too narrow for you to fit through.

Output

For each test case print out `YES` if you discover the maze to be "escape"-able, or `NO` if there is no way out.

Example Input File

```
3
3
...
.##
.#.
5
.###.
...#.
.###.
...##
.#...
10
.....#..##
#.#..###.#
#...#.#...
.#.##.##..
....#..##.
..#....#..
.####.#.#.
#.#.#.##..
.........#
.#.....#..
```

Example Output to Screen

```
NO
YES
YES
```